

**REMARKS**

Claims 1-25 are pending. Claims 24 and 25 have been amended.

**Summary of Telephonic Interview**

The undersigned wishes to thank Examiner Vo for taking the time to conduct a telephonic interview on August 10, 2007. During the interview, the undersigned presented the arguments set forth below with respect to the Section 101 rejection and the Section 102 rejection. Although no agreement was reached, the Examiner indicated that he would consider this written response.

**Rejection under 35 U.S.C. §101**

Claims 24 and 25 were rejected as being directed to non-statutory subject matter. Claims 24 and 25 have been amended to recite a “computer-readable storage medium”, which satisfies the useful, concrete, and tangible requirements under 35 U.S.C. §101. The applicants submit that, as amended, these claims satisfy the requirements under 35 U.S.C. §101. Reconsideration of the Section 101 rejection of these claims is respectfully requested.

**Rejections under 35 U.S.C. §102**

Claims 1-25 stand rejected under 35 U.S.C. 102(e) as being anticipated by Lin et al. (US 6,675,159). These rejections are respectfully traversed.

The present invention is directed to a system and method for verifying the integrity of a full-text index by checking that various components used to generate the full-text index remain consistent after the full-text index is built. As explained in the specification, generating a full-text index on a computer system can require the assistance of a variety of different software components, such as protocol components, filtering components, stemmers and wordbreakers. It is clear from paragraphs 0004-0008, 0033 and 0035 that this is what is meant by the term “components” in the claims. After the full-text index is built, those same components may be used when the index is queried. Whenever one of these software components is changed, such as during an upgrade to a newer version of the software

component or when the full-text index is moved to a different computer that has a different version of the software component, problems can arise because the version of the component used to query the full-text index may no longer be the same version used to build the index, or the component may be completely missing. When such a version mismatch exists, or when a component is removed from the computer system altogether, subsequent queries of the full-text index may not generate the same results. Consequently, whenever such a change to a software component occurs, it is often necessary to rebuild the full-text index. However, rebuilding a full-text index can be time consuming. The method and system of the present invention is intended to provide a way to relatively quickly determine whether a full-text index should be regenerated as a result of a change to one or more of the software components involved in building the index. (Spec, ¶¶ 0004 – 0008)

According to the invention, as each component needed to build a full-text index is executed, an entry is made in a component list that identifies at least the version of that component. Once entries for all components used to build the full-text index have been added to the list, the list remains associated with the full-text index. Later, the list of components can be compared to another list, such as a computer system registry, to make sure that all of the components used to build the full-text index, as well as the particular versions of those components, are still available on the computer system. If not, a warning or error can be generated to alert a user or administrator that the full-text index may need to be rebuilt. These features of the invention are recited in each of claims 1, 11, and 20. For example, claim 1 recites:

generating a list of *components used to build* a full-text index,  
the list comprising at least one component list entry, the at least  
one component list entry comprising a *version of a component*  
*used to build the full-text index* . . . .

Claims 11 and 20 similarly recite this feature. The applicants respectfully submit that Lin et al. does not teach or suggest this feature.

The Office Action cites to column 25, line 37 through column 26, line 48 of Lin et al. as teaching the claimed component list generation feature. However, while that portion of Lin et al., as well as various other cited portions of columns 11, 24 and 27 of Lin et al., do describe the use of various components used in document indexing, such as a “search collector,” a “Bayes classifier,” a “parser,” and a “sentence lexer,” nowhere do the cited

**DOCKET NO.:** MSFT-2859/306238.01  
**Application No.:** 10/743,153  
**Office Action Dated:** 05/10/2007

**PATENT**

portions of Lin et al. teach or suggest generating “a list of [those] components used to build [the] full-text index” where entries in the list identify “a version of a component used to build the full-text index.”

Because Lin et al. fails to teach or suggest the claimed component list generation feature, the applicants respectfully submit that Lin et al. does not anticipate, nor render obvious, any of claims 1, 11 or 20. Inasmuch as the remaining claims all depend either directly or indirectly from one of the independent claims 1, 11 and 20, the applicants submit that they too are patentable for the same reasons. Reconsideration of the Section 102(e) rejection of claims 1-25 is therefore respectfully requested.

### **CONCLUSION**

For the foregoing reasons, Applicants respectfully submit that the instant application is in condition for allowance.

Respectfully submitted,

Date: August 10, 2007

/Steven B. Samuels/  
Steven B. Samuels  
Registration No. 37,711

Woodcock Washburn LLP  
Cira Centre  
2929 Arch Street, 12th Floor  
Philadelphia, PA 19104-2891  
Telephone: (215) 568-3100  
Facsimile: (215) 568-3439